Claims

What is claimed is:

1. A method of managing thread pools of a computing environment, said method comprising:

3   receiving from a first requester of said computing
4   environment a request to be processed, said request
5   waiting on a response from a second requester of said
6   computing environment, and wherein said response is to
7   be serviced by a thread pool selected from a set of one
8   or more eligible thread pools; and

9   dynamically altering said set of one or more
10   eligible thread pools to provide an altered thread pool
11   set of one or more eligible thread pools, wherein a
12   thread pool of the altered thread pool set is to
13   service said response.

1   2. The method of claim 1, wherein said dynamically
2 altering comprises setting a pool mask associated with said
3 response to indicate said one or more eligible thread pools.

3. The method of claim 2, wherein said pool mask is
2 included within a data structure associated with said
3 response.

1   4.   The method of claim 1, wherein said dynamically
2   altering is initiated when it is determined that said
3   request is waiting for said response.

1   5.   The method of claim 1, further comprising selecting
2   said thread pool from said altered thread pool set to
3   service said response.

1   6.   The method of claim 5, wherein said selecting
2   comprises:

3   determining which, if any, thread pools of said
4   altered thread pool set were not included in said set
5   of one or more eligible thread pools, thereby providing
6   one or more new thread pools; and

7   selecting said thread pool from said one or more
8   new thread pools.

1   7.   The method of claim 1, further comprising
2   dispatching said response on a thread of said thread pool.

1   8.   The method of claim 7, further comprising:

2   determining whether said thread pool is
3   appropriate for the dispatched response; and

4   redispatching said response onto another thread
5   pool when the thread pool is inappropriate.

9.   The method of claim 1, further comprising
dynamically re-altering said altered thread pool set to
service one or more other responses or one or more other
4   requests.

1   10.   The method of claim 9, wherein said dynamically
2   re-altering is performed when there are no outstanding
3   callbacks to be responded to by said second requester.

1   11.   The method of claim 10, further comprising
2   determining whether there are any outstanding callbacks,
3   said determining referencing a data structure associated
4   with said second requester.

1   12.   The method of claim 1, further comprising
2   selecting a thread pool to service said request from a
3   request set of one or more eligible thread pools, wherein
4   said selecting is based on an ordering of said one or more
5   eligible thread pools.

1   13. The method of claim 12, wherein said ordering

2   comprises having a primary thread pool selectable before any

3   secondary thread pool.

1   14. The method of claim 1, wherein said receiving

2   comprises receiving said request by a server of said

3   computing environment, and wherein said first requester is a

4   first client and said second requester is a second client.

1   15. The method of claim 14, wherein said server is a

2   file server.

1   16. The method of claim 14, wherein at least one of

2   said first client and said second client runs on a same

3   physical computer of said computing environment as said

4   server.

1   17. The method of claim 14, wherein at least one of

2   said first client and said second client runs on a different

3   physical computer of said computing environment than said

4   server.

1   18. The method of claim 1, wherein said dynamically

2   altering is performed by a server of said computing

3   environment.

1   19. The method of claim 1, wherein said thread pool of

2   said altered thread pool set is to service said response to

3   avoid a deadlock with said request awaiting said response.

1   20.   The method of claim 1, wherein said dynamically
2   altering comprises receiving no indication from said second
3   requester of which thread pools are to be included in said
4   altered thread pool set.

1   21.   The method of claim 1, wherein said first
2   requester and said second requester are the same requester.

1   22.   The method of claim 1, wherein said first
2   requester and said second requester are different
3   requesters.

1   23.   A method of managing thread pools of a computing
2   environment, said method comprising:

3   dynamically determining which thread pool of a
4   plurality of thread pools is to be used to process a
5   request; and

6   processing said request using a thread of said
7   thread pool.

1   24.   The method of claim 23, wherein said dynamically
2   determining is performed when said request is a callback
3   response to another request.

1   25.   The method of claim 23, wherein said computing
2   environment comprises at least one server and at least one
3   client, and wherein said request is issued by a client of
4   said at least one client and received by a server of said at
5   least one server, and wherein said dynamically determining
6   is performed by said server.

1   26.   The method of claim 25, wherein said dynamically
2   determining comprises receiving no indication from said
3   client of which thread pool is to be used.

27.   A system of managing thread pools of a computing
environment, said system comprising:

3   means for receiving from a first requester of said
4   computing environment a request to be processed, said
5   request waiting on a response from a second requester
6   of said computing environment, and wherein said
7   response is to be serviced by a thread pool selected
8   from a set of one or more eligible thread pools; and

9   means for dynamically altering said set of one or
10   more eligible thread pools to provide an altered thread
11   pool set of one or more eligible thread pools, wherein
12   a thread pool of the altered thread pool set is to
13   service said response.

1   28.   The system of claim 27, wherein said means for
2   dynamically altering comprises means for setting a pool mask
3   associated with said response to indicate said one or more
4   eligible thread pools.

        29.   The system of claim 28, wherein said pool mask is
2   included within a data structure associated with said
3   response.

30.   The system of claim 27, wherein the dynamically altering is initiated when it is determined that said request is waiting for said response.

31.   The system of claim 27, further comprising means for selecting said thread pool from said altered thread pool set to service said response.

32.   The system of claim 31, wherein said means for selecting comprises:

means for determining which, if any, thread pools of said altered thread pool set were not included in said set of one or more eligible thread pools, thereby providing one or more new thread pools; and

means for selecting said thread pool from said one or more new thread pools.

33.   The system of claim 27, further comprising means for dispatching said response on a thread of said thread pool.

1   34. The system of claim 33, further comprising:

2   means for determining whether said thread pool is
3   appropriate for the dispatched response; and

4   means for redispatching said response onto another
5   thread pool when the thread pool is inappropriate.

1   35. The system of claim 27, further comprising means
2   for dynamically re-altering said altered thread pool set to
3   service one or more other responses or one or more other
4   requests.

1   36. The system of claim 35, wherein said means for
2   dynamically re-altering is performed when there are no
3   outstanding callbacks to be responded to by said second
4   requester.

1   37. The system of claim 36, further comprising means
2   for determining whether there are any outstanding callbacks,
3   said determining referencing a data structure associated
4   with said second requester.

1   38. The system of claim 27, further comprising means
2   for selecting a thread pool to service said request from a
3   request set of one or more eligible thread pools, wherein
4   the selecting is based on an ordering of said one or more
5   eligible thread pools.

1   39.   The system of claim 38, wherein the ordering
2   comprises having a primary thread pool selectable before any
3   secondary thread pool.

1   40.   The system of claim 27, wherein said means for
2   receiving comprises means for receiving said request by a
3   server of said computing environment, and wherein said first
4   requester is a first client and said second requester is a
5   second client.

1   41.   The system of claim 40, wherein said server is a
2   file server.

1   42.   The system of claim 40, wherein at least one of
2   said first client and said second client runs on a same
3   physical computer of said computing environment as said
4   server.

1   43.   The system of claim 40, wherein at least one of
2   said first client and said second client runs on a different
3   physical computer of said computing environment than said
4   server.

1   44.   The system of claim 27, wherein the dynamically
2   altering is performed by a server of said computing
3   environment.

1   45.   The system of claim 27, wherein said thread pool
2 of said altered thread pool set is to service said response
3 to avoid a deadlock with said request awaiting said
4 response.

1   46. The system of claim 27, wherein said means for
2 dynamically altering receives no indication from said second
3 requester of which thread pools are to be included in said
4 altered thread pool set.

1   47.   The system of claim 27, wherein said first
2 requester and said second requester are the same requester.

1   48.   The system of claim 27, wherein said first
2 requester and said second requester are different
3 requesters.

1   49. A system of managing thread pools of a computing
2 environment, said system comprising:

3   means for dynamically determining which thread
4   pool of a plurality of thread pools is to be used to
5   process a request; and

6   means for processing said request using a thread
7   of said thread pool.

1   50. The system of claim 49, wherein the dynamically
2 determining is performed when said request is a callback
3 response to another request.

1   51. The system of claim 49, wherein said computing
2 environment comprises at least one server and at least one
3 client, and wherein said request is issued by a client of
4 said at least one client and received by a server of said at
5 least one server, and wherein the dynamically determining is
6 performed by said server.

1   52. The system of claim 51, wherein said means for
2 dynamically determining receives no indication from said
3 client of which thread pool is to be used.

53.   A system of managing thread pools of a computing environment, said system comprising:

3   a processor adapted to receive from a first client
4   of said computing environment a request to be
5   processed, said request waiting on a response from a
6   second client of said computing environment, and
7   wherein said response is to be serviced by a thread
8   pool selected from a set of one or more eligible thread
9   pools; and

10   said processor being adapted to dynamically alter
11   said set of one or more eligible thread pools to
12   provide an altered thread pool set of one or more
13   eligible thread pools, wherein a thread pool of the
14   altered thread pool set is to service said response.

1   54.   The system of claim 53, wherein said processor
2   comprises a server of said computing environment.

1   55.   The system of claim 53, wherein said first client
2   and said second client are the same client.

1   56.   The system of claim 53, wherein said first client
2   and said second client are different clients.

1   57.   A system of managing thread pools of a computing
2   environment, said system comprising:

3   a processor adapted to dynamically determine which
4   thread pool of a plurality of thread pools is to be
5   used to process a request; and

6   said processor being adapted to process said
7   request using a thread of said thread pool.

58.   At least one program storage device readable by a
machine, tangibly embodying at least one program of
instructions executable by the machine to perform a method
of managing thread pools of a computing environment, said
method comprising:

    receiving from a first requester of said computing
    environment a request to be processed, said request
    waiting on a response from a second requester of said
    computing environment, and wherein said response is to
    be serviced by a thread pool selected from a set of one
    or more eligible thread pools; and

    dynamically altering said set of one or more
    eligible thread pools to provide an altered thread pool
    set of one or more eligible thread pools, wherein a
    thread pool of the altered thread pool set is to
    service said response.

59.   The at least one program storage device of claim
58, wherein said dynamically altering comprises setting a
pool mask associated with said response to indicate said one
or more eligible thread pools.

60.   The at least one program storage device of claim
59, wherein said pool mask is included within a data
structure associated with said response.

1   61. The at least one program storage device of claim
2   58, wherein said dynamically altering is initiated when it
3   is determined that said request is waiting for said
4   response.

1   62. The at least one program storage device of claim
2   58, wherein said method further comprises selecting said
3   thread pool from said altered thread pool set to service
4   said response.

1   63. The at least one program storage device of claim
2   62, wherein said selecting comprises:

3   determining which, if any, thread pools of said
4   altered thread pool set were not included in said set
5   of one or more eligible thread pools, thereby providing
6   one or more new thread pools; and

7   selecting said thread pool from said one or more
8   new thread pools.

1   64. The at least one program storage device of claim
2   58, wherein said method further comprises dispatching said
3   response on a thread of said thread pool.

1   65. The at least one program storage device of claim
2   64, wherein said method further comprises:

3   determining whether said thread pool is
4   appropriate for the dispatched response; and

5   redispatching said response onto another thread
6   pool when the thread pool is inappropriate.

1   66. The at least one program storage device of claim
2   58, wherein said method further comprises dynamically re-
3   altering said altered thread pool set to service one or more
4   other responses or one or more other requests.

1   67. The at least one program storage device of claim
2   66, wherein said dynamically re-altering is performed when
3   there are no outstanding callbacks to be responded to by
4   said second requester.

1   68. The at least one program storage device of claim
2   67, wherein said method further comprises determining
3   whether there are any outstanding callbacks, said
4   determining referencing a data structure associated with
5   said second requester.

1   69.   The at least one program storage device of claim
2   58, wherein said method further comprises selecting a thread
3   pool to service said request from a request set of one or
4   more eligible thread pools, wherein said selecting is based
5   on an ordering of said one or more eligible thread pools.

1   70.   The at least one program storage device of claim
    69, wherein said ordering comprises having a primary thread
3   pool selectable before any secondary thread pool.

1   71.   The at least one program storage device of claim
2   58, wherein said receiving comprises receiving said request
3   by a server of said computing environment, and wherein said
4   first requester is a first client and said second requester
5   is a second client.

1   72.   The at least one program storage device of claim
2   71, wherein said server is a file server.

1   73.   The at least one program storage device of claim
2   71, wherein at least one of said first client and said
3   second client runs on a same physical computer of said
4   computing environment as said server.

1   74.   The at least one program storage device of claim
2   71, wherein at least one of said first client and said
3   second client runs on a different physical computer of said
4   computing environment than said server.

1   75.   The at least one program storage device of claim
2 58, wherein said dynamically altering is performed by a
3 server of said computing environment.

1   76.   The at least one program storage device of claim
2 58, wherein said thread pool of said altered thread pool set
3 is to service said response to avoid a deadlock with said
4 request awaiting said response.

1   77.   The at least one program storage device of claim
2 58, wherein said dynamically altering comprises receiving no
3 indication from said second requester of which thread pools
4 are to be included in said altered thread pool set.

1   78.   The at least one program storage device of claim
2 58, wherein said first requester and said second requester
3 are the same requester.

1   79.   The at least one program storage device of claim
2 58, wherein said first requester and said second requester
3 are different requesters.

1   80. An article of manufacture, comprising:

2   at least one computer usable medium having

3 computer readable program code means embodied therein

4 for causing the managing of thread pools of a computing

5 environment, the computer readable program code means

6 in said article of manufacture comprising:

7   computer readable program code means for

8   causing a computer to dynamically determine which

9   thread pool of a plurality of thread pools is to

10   be used to process a request; and

11   computer readable program code means for

12   causing a computer to process said request using a

13   thread of said thread pool.

1   81. The article of manufacture of claim 80, wherein

2 the dynamically determining is performed when said request

3 is a callback response to another request.

1   82. The article of manufacture of claim 80, wherein

2 said computing environment comprises at least one server and

3 at least one client, and wherein said request is issued by a

4 client of said at least one client and received by a server

5 of said at least one server, and wherein the dynamically

6 determining is performed by said server.

1   83. The article of manufacture of claim 82, wherein

2 the dynamically determining comprises receiving no

3 indication from said client of which thread pool is to be

4 used.

<p style="text-align:center">* * * * *</p>